AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented method for examining an inlined function using a performance analysis tool, said method comprising:

identifying an inlined function in ~~computer code~~ source code for a binary executable;

inserting a breakpoint at the start of said inlined function in said binary executable; and

replacing said inlined function with a long branch to a shared memory probe code sequence.

2. (Original) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 further comprising:

creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

3. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said step of using a performance analysis tool to perform a desired task comprises:

using said performance analysis tool to perform instrumentation on said inlined function.

4. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said step of using a performance analysis tool to perform a desired task comprises:

using said performance analysis tool to perform mapping of samples to said inlined function.

5. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said performance analysis tool is comprised of an instrumentation application.

6. (Previously Presented) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said performance analysis tool is comprised of a sampling application.

7. (Currently Amended) A computer-readable medium embodying instructions that cause a computer to perform a method for examining an inlined function using a performance analysis tool, said method comprising:

identifying an inlined function in source code for a binary executable;

inserting a breakpoint at the start of said inlined function in said binary executable; and

replacing said inlined function with a long branch to a shared memory probe code sequence.

8. (Previously Presented) The computer-readable medium of Claim 7 further comprising instructions that cause said computer to perform said method further comprising:

creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

9. (Previously Presented) The computer-readable medium of Claim 7 wherein said step of using a performance analysis tool to perform a desired task comprises:

using said performance analysis tool to perform instrumentation on said inlined function.

10. (Previously Presented) The computer-readable medium of Claim 7 wherein said step of using a performance analysis tool to perform a desired task comprises:

using said performance analysis tool to perform mapping of samples to said inlined function.

11. (Previously Presented) The computer-readable medium of Claim 7 wherein said performance analysis tool is comprised of an instrumentation application.

12. (Previously Presented) The computer-readable medium of Claim 7 wherein said performance analysis tool is comprised of a sampling application.

13. (Currently Amended) An apparatus for examining an inlined function using a performance analysis tool, said apparatus comprising:

means for identifying an inlined function in source code for a binary executable;

means for inserting a breakpoint at the start of said inlined function in said binary executable; and

means for replacing said inlined function with a long branch to a shared memory probe code sequence.

14. (Original) The apparatus of Claim 13 further comprising:

means for creating a data structure which maintains location information for said inlined function and information related to said desired task for said inlined function.

15. (Original) The apparatus of Claim 13 wherein said means for performing a desired task on said inlined function further comprises means for performing instrumentation on said inlined function.

16. (Original) The apparatus of Claim 13 wherein said means for performing a desired task on said inlined function further comprises means to perform mapping of samples to said inlined function.

17. (Previously Presented) The apparatus of Claim 13 wherein said performance analysis tool is comprised of an instrumentation application.

18. (Previously Presented) The apparatus of Claim 13 wherein said performance analysis tool is comprised of a sampling application.

19.   (Original) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

20.   (Currently Amended) The computer-implemented method for examining an inlined function using a performance analysis tool as recited in Claim 1 wherein said inserting further comprises:

reading source correlation information from within said binary executable computer code; and

obtaining start and end addresses for said inlined function using said source correlation information.

21.   (Original) The computer-readable medium of Claim 7 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

22.   (Currently Amended) The computer-readable medium of Claim 7 wherein said inserting further comprises:

reading source correlation information from within said binary executable computer code; and

obtaining start and end addresses for said inlined function using said source correlation information.

23.   (Original) The apparatus of Claim 13 wherein said shared memory probe code sequence saves registers, executes the original bundle of said inlined function, restores said registers, and jumps back to said computer code.

24.   (Currently Amended) The apparatus of Claim 13 further comprising:

means for reading source correlation information from within said binary executable ~~computer~~ code; and

means for obtaining start and end addresses for said inlined function using said source correlation information.